CSc 174 Database Management Systems

16. NoSQL

Ying Jin
Computer Science Department
California state University, Sacramento

No SQL Characteristics

- Not requiring a schema
- Any constraints on the data have to be programmed in the application programs
- Less powerful query languages
  - Typically a set of functions and operations as programming API
  - Do not provide join

Categories of NoSQL systems

- Document-based
- Key-value store
- Column-based or wide column
- Graph-based

Categories (1)

- Document-based
  - Store data in the form of documents
  - MongoDB
  - CouchDB

Categories (2)

- Key-value store
  - Every value(data itme) must be associated with a unique key
  - Retrieving the value by supplying the key must be very fast.
  - Amazon DynamoDB, Oracle NoSQL database,Redis, Apache Cassandra (also categorized as column-based)

Categories (3)

- Column-based or wide column
  - Partition a table by column into column families
  - Google BigTable
  - Apache Hbase
  - Apache Cassandra
- Graph-based
  - Neo4j, GraphBase

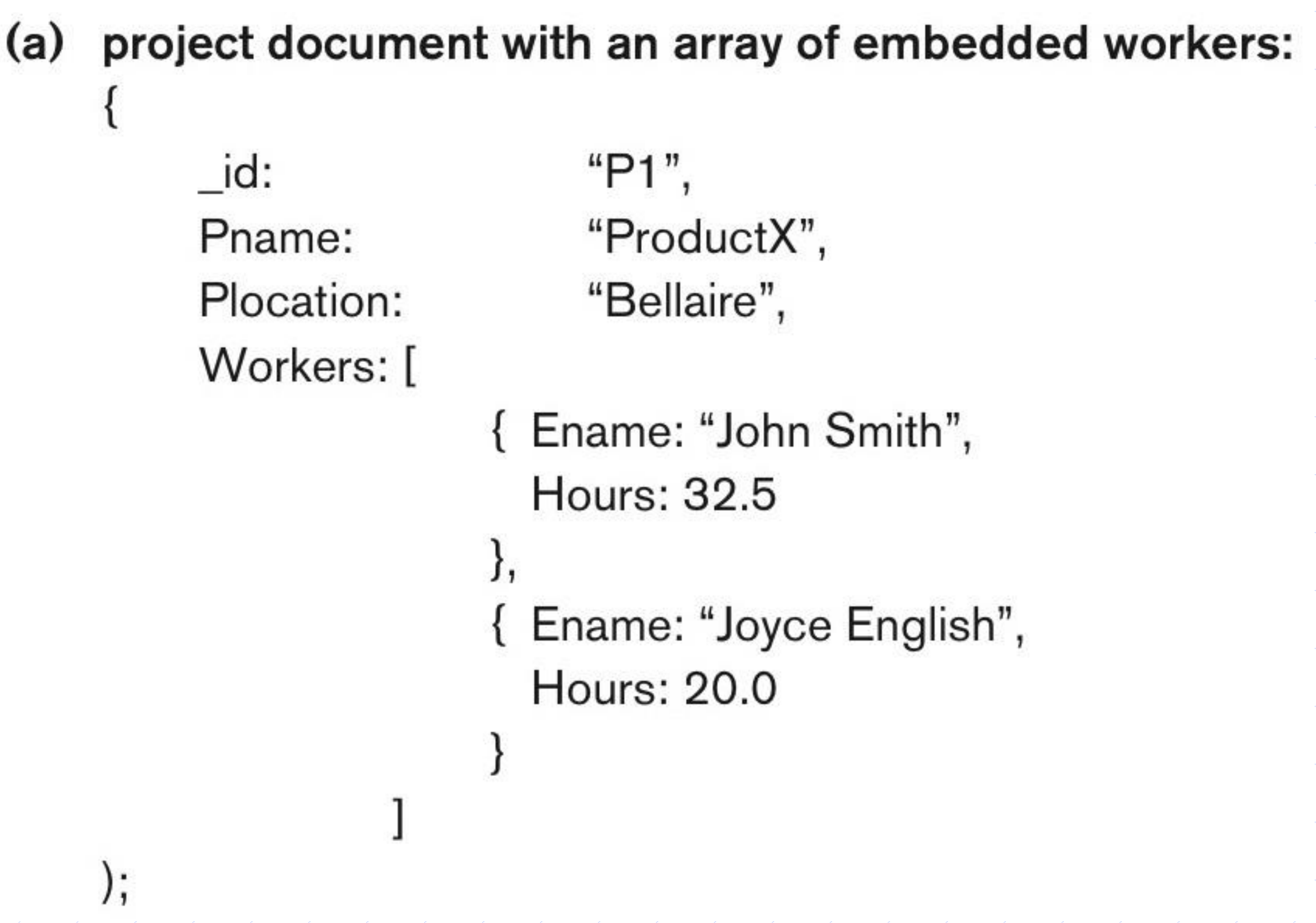
- MongoDB

MongoDB

- Document-based NoSQL
- Documents are stored in BSON (Binary JSON: Binary JavaScript Object Notation)
- Individual documents are stored in a collection

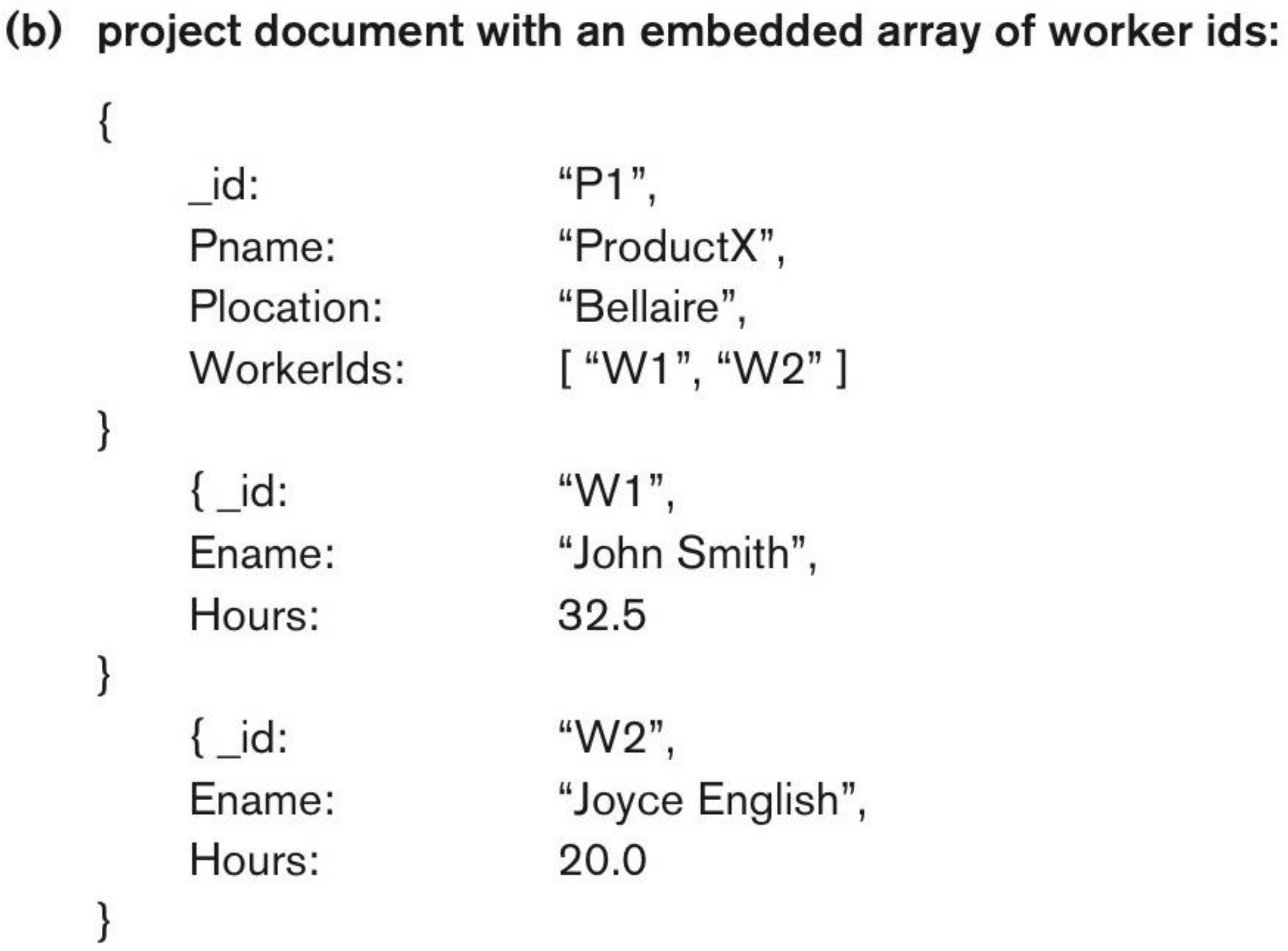
Collection

- db.createCollection(“Project”, {capped:true, size: 1310720, max:500})
- db.crateCollection(“Worker”, {capped:true, size: 5242880, max:2000}))
- Each document in a collection has a unique ObjectId, _id
  - Can be specified by users
  - If not specified by users, it is system-generated



(a) project document with an array of embedded workers:

```
{
      _id:                    "P1",
      Pname:                  "ProductX",
      Plocation:              "Bellaire",
      Workers: [
                        { Ename: "John Smith",
                          Hours: 32.5
                        },
                        { Ename: "Joyce English",
                          Hours: 20.0
                        }
                  ]
);
```



(b) project document with an embedded array of worker ids:

```
{
     _id:            "P1",
     Pname:          "ProductX",
     Plocation:      "Bellaire",
     WorkerIds:      [ "W1", "W2" ]
}
     { _id:          "W1",
     Ename:          "John Smith",
     Hours:          32.5
}
     { _id:          "W2",
     Ename:          "Joyce English",
     Hours:          20.0
}
```

CRUD Operations

- db.<collection_name>.insert(<document(s)>)
- db.<collection_name>.remove(<condition>)
- Db.<collection_name>.find(<condition>)
- Db.<collection_name>.update(<condition>)

- Neo4j

Data Model

- Node
  - Lable: nodes with the same label are grouped into a collection
- Relationship
  - Directed: with a start node and end node

Query Language -Cypher

- Create nodes
- Create relationships
- query

Query

MATCH

WITH

WHERE

RETURN

ORDER BY

LIMIT

Query Example

- A)Retrieve projects and hours per weeks that the employee with Empid=2 works on.
- B)Retrieve all employees and the projects they work on, sorted by Ename, but only returns the first 10 answers.
- C) Retrieve the employees who work on more than two projects, as well as the number of projects each employee works on.

These slides are based on the textbook:

R. Elmasri and S. Navathe, *Fundamentals of Database System*, 7th Edition, Addison-Wesley.